Enhancement of documents and information management by researchers

Abdel Hamid, Boujdad Mkadem

Abstract

This paper presents a report of an investigation of contemporary software tools that assist the information seekers and users; it also presents a document and information management system based on existing software. This system is intended for research purposes at personal and group level. It should make searching, managing and retrieving documents more complete than they are compared with other document retrieval systems. All this in a given situation: students and researchers dealing with PDF documents downloaded from electronic journals collections.

Introduction

We think that a system which is able to help manage and automate annotation of documents using a classification system [Koraljka, 2003] and at the same time helping him manage all other documents would ease the work of a researcher. Since most of the documents on these databases are in PDF; the research has gone towards tools that present high performances in searching and managing them. Text retrieval software are widely used on desktops now (Google Desktop Search, Copernic Desktop Search for instance), but are very few in term of efficiency. We had tested several software packages, and we have come to the conclusion that document retrieval tools are limited as a researcher needs more than retrieving rapidly a document after a successful search. Search tools combined with management components may well be more interesting in document and information management.

Problem statement

The student community at the Vrije Universiteit Brussel is afforded a wide range of electronic journals. Most of these article databases collections are documents in PDF format. Once students download these documents on their desktops; their use and the management are ignored. At the end they find themselves with big collections of documents in their desktops and with trouble retrieving them once they need to find (or re-find). Our main focus at that level was on software that *could be able* to do fulltext search on PDF documents, and *preferably be* open source or at a lesser degree be of an acceptable price.

Methods

A series of tests have been driven. We tested open source and free software like *Docsearcher* [Docsearcher] and *Windows Desktop Search* [Windows Desktop].The latter lacks a filter for PDF files. The Adobe Company does offer that adequate filter (*Ifilter*). It's an efficient way to make the WDS work perfectly. The two of them did wok admirably well.

Many other systems were tested but we selected only those listed below as they were in our opinion the most relevant. Some of these software packages are excellent but had one (or more at the same time) of the following disadvantages:

- too expensive,
- lack annotating and document.

These tests were all done on a Pentium III/ 256 RAM machine, between 2003 and 2005. We focused mainly on the following criteria:

- executing full text search within PDF format documents
- multiplatform or Working on windows machines (widely used)
- speed of indexing and creating the initial index
- search saving and document type filtering are not as important but give an idea about the quality of the system..

Results and discussions

We have found that the most convenient ones (because they satisfy the main condition of performing PDF fulltext search) with regards to our needs were: *Google Desktop Search* and *Copernic Desktop Search* from Google™ and Copernic™ respectively

Name of the software	Files types			Platform	Indexing	Search features		
	Pdf	Html	Other		Fast/Slow	Search saving	Full text	Type filtering
Dtsearch 6.30	●	●	●	Windows	35 min	●	●	●
Copernic Desktop Search 1.63	●	●	●	Windows	Real time	○	●	●
Effective File Search 3 07		●	●	Windows	○	○	○	●
80-20 Retriever 206 SP1		●	●	Windows	No	●	○	●
windows Desktop Search with PDF Ifilter v6.0	●	●	●	Windows	○	○	●	
Google beta	●	●	●	Windws/other	Real time	○	●	●
Docsearcher 3.87	●	●	●	All (java)	Real time	○	●	●
KIM	●	●	●	All (java)	On demand	Not applicable	●	○

Figure 1 *Retrieval software features comparison*

We realised that our investigation won't go any farther in helping the researcher with regard to the main objective: help manage documents and generally all information useful for a researcher. Then we tried to find other tools that would meet our requirements.

On his own pc, a researcher is confronted to the same problem finding and re-finding documents already retrieved. We decided to try to go further and see what could be done at management level. Some tools do exist but they are less known than *Google* or *Copernic*. We think these tools may really help the researcher in his work. There is a need for more than good retrieval software as the amount of documents is increasing.

We have come to the certainty that we really need to find a federating tool, one that could represent a real enhancement with regards to what we tested in our overview. Although this tool doesn't really exist as it is but we think that we could afford the ground rules and

method to achieve the task of constructing (or federating it). This is why we thought about a prototype similar to *KIM*.

Born in Ontotext Labs, the KIM (Knowledge and Information Management) is different "from the classical IR task: documents are retrieved based on relevance to NEs instead of words" [Ontotext] NEs which stands for "Named entities".

We recommend the ONTOTEXT KIM [Ontotext] software for the purpose outlined above. This software is in our opinion what we need for the following reasons:

- It is free for research purposes.
- It is multiplatform (KIM is written in java).

KIM is a tool that can visualise documents and annotate them with great ease using ontologies or thesauri. What is really interesting as a feature is that a researcher can define his own ontology. It is very important as a researcher may define an ontology with a group of persons. This is ideal when a team is working in a project: people with the same lexical agreement are expected to be more efficient in finding relevant document that were stored and indexed and/or annotated by others.

This system is intended to make the researchers more efficient at their work as it assists them in the burden of document and information management. We are then in a crossroad as we need to know if searching document were made easy on not? Is research well assisted by this solution or not?

KIM is not a database. It's a combination of servers (*KIM*, *Sesame* and *Tomcat*) for creating ontologies in order to help annotating the document using a plugin (KIM plugin) for the IE browser. It extracts the keywords automatically (only those previously defined in the ontologies) stores them for further search. What are extracted are not verbs; only "entities" or concepts of noun type. It does perform an automatic linking while a new document is added applying the language processing tools loaded in the system. It makes available the information about concepts but also about persons. The importance of contact persons and organisations in the research field is highlighted by Hertzum [Hertzum, 2000]. KIM may be adapted with researcher's lexical fields: creating their own ontologies with the relation they see possible between the concepts related to their field of research.

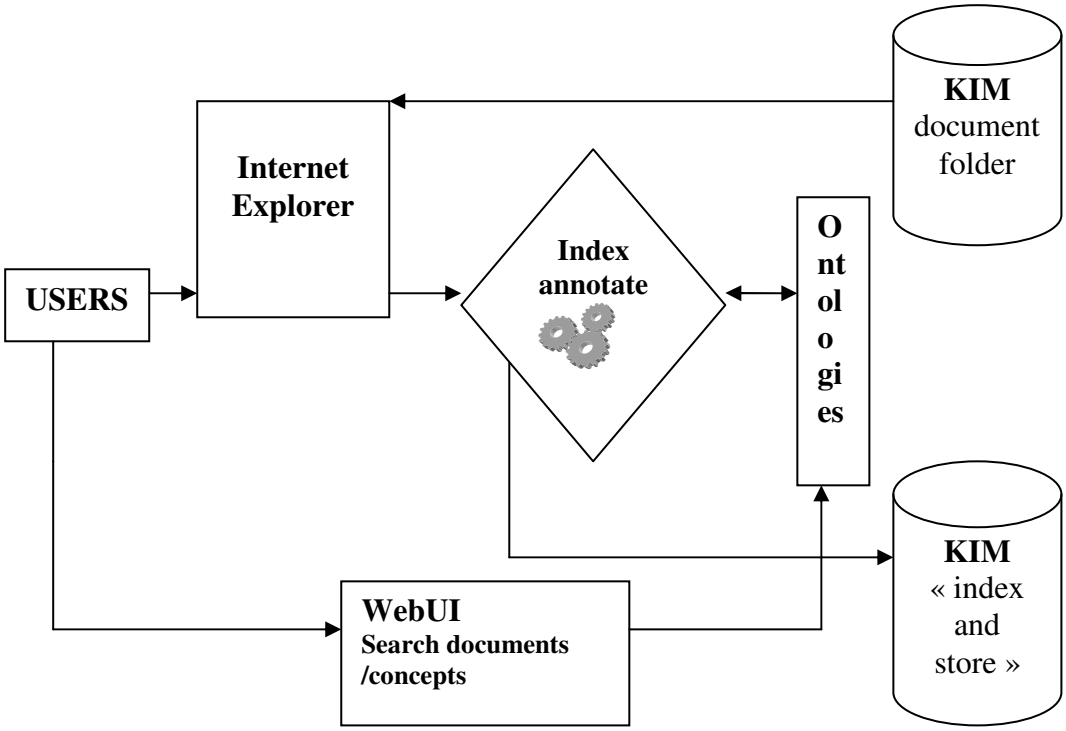

Figure 2 The KIM system

The idea of using annotations for retrieval purposes is not new but it still attracts research. Some other studies [Maristella, 2005] consider the annotations as discussion: Usenet is an example; they consider them as having interesting impact on search efficiency.

Technically two solutions are possible: the first one is a JavaScript page that calls the execution of some tasks. The first plugin should execute the load and conversion of PDF documents and converts them into text documents via the *BCldrake* [BCLDRAKE] or *Jade plugin*. Then it launches KIM after storing the files converted in the Corpus folder within KIM (see figure3).

The second possibility is to insert directly into the *Internet Explorer* a plugin that appears as an icon that would launch the execution of complementary tools : Jade [JADE] or *BCLdrake* for instance. Once the conversion is done we can store the converted files in the KIM Corpus folder.

At this level, we should be able to say that researchers can with few mouse clicks:

- create their own ontology/thesauri at personal or/and group level
- index/annotate their files using the ontologies of their choice
- retrieve them using different search criteria: personalised annotations fields, keywords…etc, using the *WebUI* component.
- see the ranking of the documents: the most highlighted passage or document is probably very interesting.

A similar project is “Gate” [GATE] it comes along with the KIM which uses it). It is “an ontology based semantic annotation of web mined documents”. Like KIM its core concept is what we call Information extraction and not Information retrieval as it’s focused on analysing texts and “presents only the specific information from them that the user is interested in.” [Cunningham]

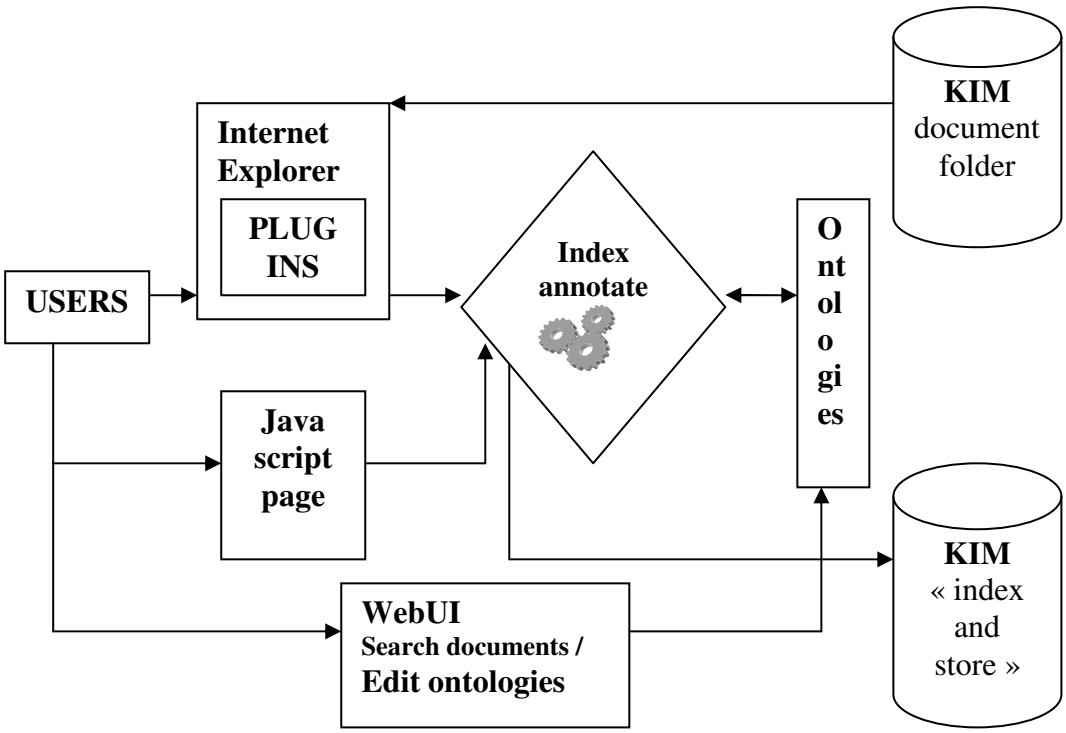

Figure 3 The KIM system with the recommended new features

Conclusion

This work had begun with an overview of text retrieval software that is capable of performing fulltext search on PDF document. This task has evolved in the direction of using annotations as a context subject to enhancing the accuracy of the recall. This has led us towards a document and information management system using ontologies in the form of annotations as classifying terms. These annotations can represent scientific concepts as well as persons or their positions and organisations.

The idea inspired from the Ontotext KIM, may well represent a good and acceptable solution within the researchers world. This tool automates annotating and can greatly help in enhancing retrieval and management of documents by increasing the accuracy of search and the clustering of documents. This system may easily be generalised to meet advanced facilities. Intelligent agent may use the ontologies stored in the personal computer and performs all the previously described asks on documents using profiles and annotations that the user himself adds to the system.

References:

1. [Koraljka 2003] Koraljka, G. (2003) Using controlled vocabularies in automated subject classification of textual Web pages, for browsing. [Online]. Retrieved October 25, 2005, from www.dei.ist.utl.pt/~jlb/ECDL2005-DC/03-KoraljkaGolub/03-Golub-final.doc

2. Maristella Agosti and Nicola Ferro. "Annotations as context for searching documents. Information Context" in Fabio Cretani and Ian Ruthven,editors, Nature, Impact, and Role: 5th International Conference on Conceptions of Library and Information Sciences, CoLIS 2005, Glasgow, UK, June 4-8, 2005. Proceedings

3**.** [Cunningham] Diana., Maynard, Milena Yankova, Niraj, Aswani, and Hammish Cunningham, (2003) Automatic creation and monitoring of semantic metadata in a dynamic knowledge portal, [Online], retrieved April 29,2005, from http://gate.ac.uk/sale/aimsa04/aimsa.pdf

4. [Cunningham] Cunningham, H. 2003, Information Extraction, automatic, [Online], retrieved March 13, 2005, from http://gate.ac.uk/sale/ell2/ie/main.pdf

5. [Hertzum] Morten Hertzum and Annelise Mark Pejitersen, "the information seeking practices of engineers; searching for documents as well as for people", Information processing and management 36 (2000) p.761-778.

6. [GATE] http://gate.ac.uk

6. [Kiryakov2003] Atanas Kiryakov, Borislav Popov, Damyan Ognyanoff, Dimitar Manov, Angel Kirilov, Miroslav Goranov, "Semantic Annotation, Indexing, and Retrieval", [Online], retrieved February ,10, 2005, from http://www.ontotext.com/publications/SemAIR_ISWC169.pdf

7. [Docsearcher] A java open source search tool http://docsearcher.sourceforge.net/

8. [Windows Desktop] the Windows Desktop search engine. 2005 http://www.microsoft.com/windows/desktopsearch/enterprise/default.mspx

9. [Ontotext] http://www.ontotext.com

10. [JADE] http://www.bcltechnologies.com/document/products/jade/jade.htm

11. [BCLDRAKE] http://bcldrake.say-it-now.com/

12. [Scott, 2005] Scott, D (2005) Deep file divers [Electronic version], *PC World Magazine* http://www.pcworld.com/reviews/article/0,aid,122096,00.asp